

# The Virtual Windtunnel: A High-Performance Virtual Reality Application

Steve Bryson, Computer Sciences Corporation  
Applied Research Branch, Numerical Aerodynamic Simulation Systems Division  
NASA Ames Research Center, Moffett Field, Ca. 94035-1000  
bryson@nas.nasa.gov

## Abstract

This paper describes work in progress on a virtual environment designed for the visualization of pre-computed fluid flows. The overall problems involved in the visualization of fluid flow are summarized, including computational, data management, and interface issues. Requirements for a flow visualization are summarized. Many aspects of the implementation of the virtual windtunnel were uniquely determined by these requirements. The user interface is described in detail.

## 1. Introduction

The virtual windtunnel [1][2] is the application of virtual reality interface techniques to the problem of the visualization of the results of computational fluid dynamics (CFD) computations. These results are typically vector and scalar fields in three-dimensional space which change over time. CFD datasets are typically extremely complex, involving time and space-varying structures such as vortices, recirculation, and oscillation.

It was expected that the three-dimensional display and control offered by virtual environment systems would greatly facilitate the investigation of fluid flow data sets, allowing the researcher to explore the data directly. While these expectation have been met, the requirements of flow visualization and the requirements of virtual environment systems were often at odds, forcing compromises between the two sets of requirements. These compromises were critical for the success of the virtual windtunnel and are discussed in this paper. The other topic is the design of the interface which facilitates the visualization task.

The virtual windtunnel is a work in progress. During 1993 it is expected that the virtual windtunnel will be released as a tool for use by a limited user community at NASA Ames. As it is not currently in general use, we can offer only preliminary evaluations of its actual utility. Even in its preliminary stages, however, it has been widely demonstrated and has received an enthusiastic response from both the fluid research and the computer graphics communities.

## 2 Requirements for the Virtual Windtunnel

The virtual windtunnel is at the intersection of two highly demanding applications of computer graphics: real-time interactive virtual environment systems and unsteady fluid flow visualization. We shall discuss the requirements of these two fields separately.

### 2.1 Requirements for Unsteady Fluid Flow Visualization

The visualization of modern unsteady fluid flow data sets must confront the following issues:

- **Data management:** The datasets are often in the several gigabyte size range. This includes several timesteps of vector and scalar data. The data sets addressed by the virtual windtunnel are defined on several stretched,

overlapping grids per timestep. The grids are stretched to conform to the body of the aircraft around which the air is flowing.

- **Computation:** The visualization techniques, such as those based on particle integration (streamlines, streaklines, and particle paths) (figure 1) and isosurfaces, require computations using the CFD data. The computations must be sufficiently accurate to reflect flow phenomena. These computations can be considerable, and for some visualization techniques, i.e. particle paths, potentially requires unpredictable access to the entire data set.
- **Graphics:** The results of the visualization computations must be rendered with sufficient accuracy to represent the flow phenomena. Some visualization techniques, i.e. streamlines, can be rendered as simple lines. Isosurfaces, however, can contain several hundreds of thousands of polygons.

Complex flows may require several visualization displays to be operating simultaneously, further compounding the computation and graphics problem.

- **Cooperative visualization:** The fluid flow community, like any scientific community, operates by cooperative investigation of phenomena. Thus a flow visualization system should support shared, cooperative visualization.
- **User acceptance:** Flow researchers will use a system when the difficulties and training investment are outweighed by the advantages of the visualization system. This means that as much functionality as possible should be included, with no features that do not contribute to that functionality. Difficulty of use should be kept to an absolute minimum.

The benchmark data set used for the virtual windtunnel system is that of a simulated harrier aircraft in hover [3]. This data set has 106 timesteps, with 18 grids/timestep for a total of 2,833,700 points per timestep, or 56 megabytes per timestep, with a total size of 5.6 gigabytes.

## 2.2 Virtual Environment Interface Requirements

Virtual environment systems rely on an illusion of immersion in an interactive three-dimensional world. This illusion is typically attained through a head-coupled wide-field stereoscopic display combined with a three-dimensional tracker and dataglove. To sustain the illusion and allow useful interaction, the virtual scene must be rendered faster than about 8-10 frames per second. The requirement of good three-dimensional interactivity and control further demands that the time from when a user initiates an action such as movement of the hand to the time when that movement is reflected in the display should be less than 0.1 seconds. Longer times significantly impact user performance in tracking and pick and place tasks [4][5]. Thus if the user interaction controls a visualization task, as is the case in the virtual windtunnel, all computation and display involved in that visualization must take place within 0.1 seconds.

## 3 Implementation

Simultaneously meeting the requirements of large size data management, extensive computation, and extensive graphics within the virtual environment time constraints in the virtual windtunnel required careful choice of software architecture, hardware, algorithms, and interface design. This section will summarize the design choices that were made to meet these requirements.

### 3.1 Design Choices

The requirements listed in the last section were each met in different ways:

- **Data management:** The requirement that the data be accessed in apriori unpredictable ways within 0.1 sec-

onds forces the data to be resident in physical memory. No mass storage devices have sufficient bandwidth to access even a single timestep within this time constraint. Example: a single timestep of only the velocity vector field data for the harrier data set described in section 2.1 is 36 megabytes in size, requiring a bandwidth of 360 megabytes per second. When the available physical memory is not sufficient to hold the entire data set, a subset of the data must be chosen. This subset may be generated by either subsampling in time or specification of a small volume of space.

- **Computation:** Particle integration visualization techniques such as streamlines and particle paths are performed by an adaptive second-order Runge-Kutta integration algorithm. The adaptive step size is chosen so that the particle integration takes  $n$  steps in a grid cell. The choice of  $n$  is controlled in real-time by the user. The integration is performed in grid coordinates, where the coordinates represent actual indices into the data array. In this way time-consuming lookups of the current location for each point of integration using the physical position grid is avoided. The points which are the result of the integration are converted into physical coordinates for rendering via the position grid. When performing the integration, particles may move from one grid to another, invalidating the current computational coordinates (which are defined only for the current grid). Finding the computational coordinates for the new grid requires a table search to convert the current physical coordinates to the new computational coordinates. This extra computation effectively prohibits the vectorization of the particle integration, severely impacting performance on vector processors. This choice of integration method is capable of integrating several thousand particles within the 0.1 second time constraint, allowing the user to observe the paths change in real time as the sources of the integrations are moved about. The marching cubes algorithm for the computation of isosurfaces is adaptively implemented, with the user able to control the step size in computational coordinates.
- **Graphics:** The virtual scene in the virtual windtunnel contains the following: representation of an object, typically an aircraft, around which the simulated air is flowing; various visualization graphics; virtual tools such as menus and sliders; reference markers such as the hand cursor and a floor/horizon reference. Isosurface and object rendering may contain many more polygons than can be rendered within the time constraint, requiring subsampling, compromising quality of image for speed. The user has real-time control over the amount of subsampling. The graphical representation of the paths that arise from particle integration can be simple lines. These lines become a performance bottleneck when the number of integrated points approaches 10,000. The virtual tools can be turned on and off at will, avoiding scene clutter. The hand is represented by a simple three-dimensional crosshair. An articulated hand model is not used to avoid performance overhead and to avoid scene clutter.

### 3.2 Hardware

There were two hardware configurations implemented for the virtual windtunnel: stand-alone and distributed. Each configuration used the same virtual reality interface hardware. The choice of hardware architecture is primarily driven by the data management requirements. It is expected that as the physical memory and computational power of workstations increases, the stand-alone architecture will become the most useful architecture.

The stand-alone system was implemented on a single workstation, which performed the computation, managed data, handled the I/O devices, and rendered the virtual scene in stereo to the display (see figure 2). The primary workstation used is a Silicon Graphics 380 4D/VGX workstation with 8 33 MHz R3000 processors for a total computational performance of 37 megaflops and 256 megabytes of physical memory. This system has a graphics performance rated at 800,000 polygons/second. The software architecture separates the computation, rendering, and I/O collection into parallel processes using shared memory. In this way no one task slows another. This is important as the graphics must update to reflect the new position of the user's head even though new computations may not have completed. Also, collection of hand and head position can occur as fast as possible. Currently the glove data is collected at 38 Hz, while the head position is collected at 45 Hz.

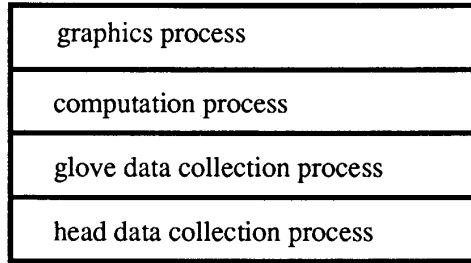


Figure 2: Parallel processes in the stand-alone architecture.

The distributed system uses a Convex C3240 computer with four vector processors and one gigabyte of physical memory for computations. Silicon Graphics VGX family workstations are used for rendering the virtual scene and handling the virtual environment interface hardware. The distributed architecture supports shared interaction, supporting two workstations with virtual environment interfaces (see figure 3). The design of the distributed architecture is greatly facilitated by the use of the Distributed Library by Michael Gerald-Yamasaki [6]. The communications between the Convex and the workstations is over the UltraNet, a gigabit network. Due to limitations with the UltraNet interface card in the workstations, the UltraNet is capable of 13 megabytes/second into the workstations. The primary motivation for the distributed architecture is the access to the gigabyte of memory. The software architecture is shown in figure 4.

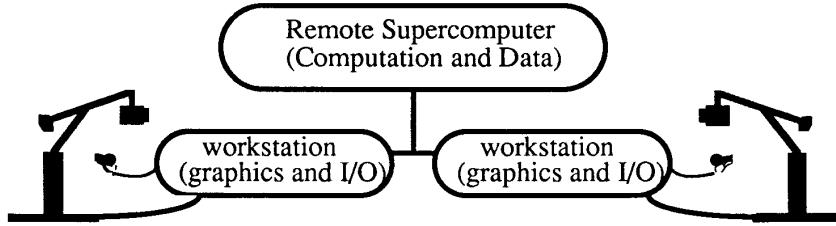


Figure 3: Distributed shared architecture of virtual windtunnel

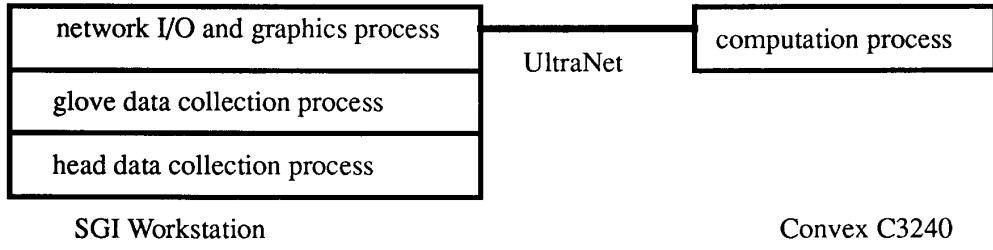


Figure 4: Software architecture in distributed version of virtual windtunnel

The choice of virtual environment display is forced by the requirement that the displayed image be of as high a resolution as possible. The resolution of LCD-based head-mounted displays was considered unacceptable for the purposes of flow visualization. The Fake Space Labs BOOM IIC, a boom-mounted head-coupled stereoscopic display (based on a system described in [7]) with

1000x1000 pixel resolution under the wide field optics was chosen because of its superior display. The BOOM IIC also has superior head-tracking capability via optical encoders at the joints of the supporting boom structure. The ease of use of the boom is also a major advantage over head-mounted systems, greatly facilitating user acceptance. The user interaction is via the standard VPL Dataglove Model II, which uses a Polhemus Isotak three-dimensional tracker for hand position and orientation, and fiber optic technology to measure the bend of finger joints. The virtual environment interface is shown in figure 5.

### 3.3 Interface

All operations in the virtual windtunnel are performed with the Dataglove interface. There are two classes of operations: direct manipulation of objects in the environment, and indirect control via virtual menus and sliders (figure 6). All operations are performed with the glove using only two gestures: grab (fist) and point.

Sources of particle integration are grouped into lines known as rakes. There can be several rakes in the environment, each of which may contain sources for one or all of the particle integration types described in section 2. The rake is moved via direct manipulation. Grabbing the center of the rake with the Dataglove causes the rake to move rigidly with the user's hand. Grabbing either end causes the end grabbed to be moved while the other end remains stationary. A sphere is drawn when the virtual hand is sufficiently close to a rake to grab part of it, providing feedback to the user (figure 7). This interface allows a rake to be positioned arbitrarily with arbitrary orientation. This method of controlling the orientation of the rakes was chosen over the use of the hand orientation due to the limited range of motion of the human wrist.

Various aspects of the environment are controlled via virtual menus[8]. Making a point gesture in empty space in the virtual environment causes a multi-level hierarchical menu to pop up in three-dimensional space within the user's field of view. The menu remains as long as the point gesture is held by the user. While the menu is up, the user's hand orientation information is used to point at various menu items. Releasing the point gesture while pointing at a menu item causes that menu item to be executed.

Rake parameters such as the number of particle integration sources are indirectly controlled via virtual sliders. These sliders exist in the three-dimensional environment and output values determined by the user making a point gesture in the active region of the slider. The sliders can be moved by making a grab gesture in the region of the slider. Sliders are toggled on and off via the virtual menus.

Navigation within the environment uses a paradigm in which the user stays in one place and moves the environment about. This is accomplished by making a grab gesture in empty space and moving the entire environment with the motion of the user's hand. When combined with a variable scale controlled via a virtual slider, this interface allows rapid and high-precision maneuvering in the environment. The virtual tools such as menus and sliders are not effected by this motion. The scale and grab paradigm is significantly easier to control than the point and fly navigation paradigm.

To summarize, these are the following actions of the gestures in the virtual windtunnel:

context \ gesture	fist	point
empty space	move data	pop up menu
virtual slider	move slider	change slider value

context \ gesture	fist	point
rake grab point	move rake	no action

The extensive use of hand gestures read by the glove in the virtual windtunnel requires a robust and reliable gesture recognition algorithm. This is accomplished using only the middle joints of the four fingers. The values measured at the knuckle joints and the thumb are ignored. First the raw values output by the Dataglove are calibrated to actual finger bend angles. Then the angles read by the glove are compared with a lookup table, to identify if the gesture is either a fist or a point. Recognizing only three gestures (fist, point, no gesture) allows forgiving and tolerant gesture recognition. With this gesture recognition algorithm, new users require a few minutes training and practice to make the gesture reliably. The glove can be calibrated from within the environment, using only two gestures. The calibration process is controlled via buttons on the BOOM IIC display.

## 4 Conclusions

The virtual windtunnel system has successfully implemented a flow visualization application in a virtual environment. While many refinements will be required to turn the virtual windtunnel into a useful tool, the basic issues have been addressed and solved.

## Acknowledgments

The virtual windtunnel has had many participants. Creon Levit and the author designed the initial concept and prototype. The distributed architecture was developed by Michael Gerald-Yamasaki and the author. Al Globus and Jeff Hultquist contributed computational code, and Rick Jacoby and Diglio Simoni contributed interface code.

## References

- [1] Bryson, S. and Levit, C., "The Virtual Wind Tunnel: An Environment for the Exploration of Three Dimensional Unsteady Flows", *Proceedings of Visualization '91 San Diego, Ca, Oct. 1991*, also *Computer Graphics and Applications* July 1992
- [2] Bryson, S. and Gerald-Yamasaki, M., "The Distributed Virtual Wind Tunnel", *Proceedings of Supercomputing '92 Minneapolis, Minn, Nov. 1992*
- [3] Smith, M., Chawla, K., and Van Dalsem, W., "Numerical Simulation of a Complete STOVL Aircraft in Ground Effect", paper AIAA-91-3293, American Institute of Aeronautics 9th Aerodynamics Conference, Baltimore Md. 1991
- [4] Bryson, S., "Impact of Lag and Frame Rate on Various Tracking Tasks", *Proceedings of SPIE Conference on Stereoscopic Displays and Applications*, San Jose, Ca., Feb. 1993
- [5] Sheridan, T. and Ferriell, W, *Man-Machine Systems*, MIT Press, Cambridge, Ma. 1974
- [6] Yamasaki, M., Distibuted Library, NAS Applied Research Technical Report RNR-90-008, April 1990
- [7] MacDowall, I., Bolas, M., Pieper, S., Fisher, S. and Humphries, J., Implementation and Integration of a Counterbalanced CRT-based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environment Applications, *Proceedings of the 1990 SPIE Conference on Stereoscopic Displays and Applications*, Santa Clara, Ca. 1990
- [8] Jacoby, R. H., Using Virtual Menus in a Virtual Environment. *Proceedings of the Symposium on Electronic Imaging Science & Technology*, International Society for Optical Engineering/Society for Imaging Science & Technology, Volume 1668.

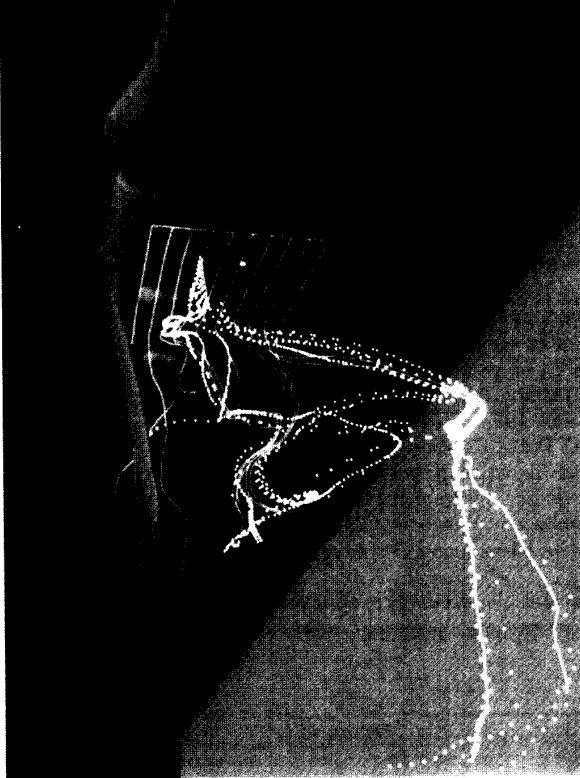


Figure 1: Streamlines and streaklines in the virtual windtunnel



Figure 5: The virtual environment interface hardware

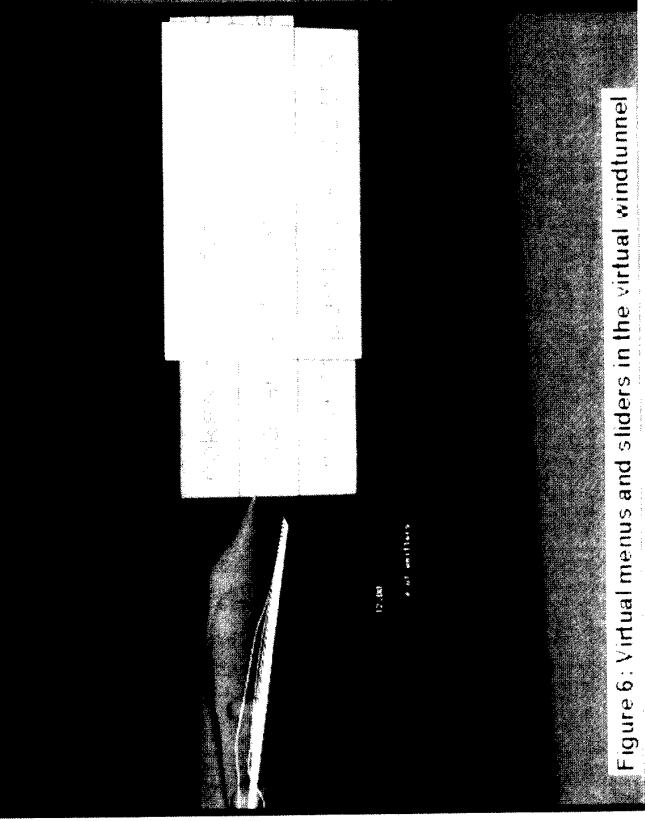


Figure 6: Virtual menus and sliders in the virtual windtunnel

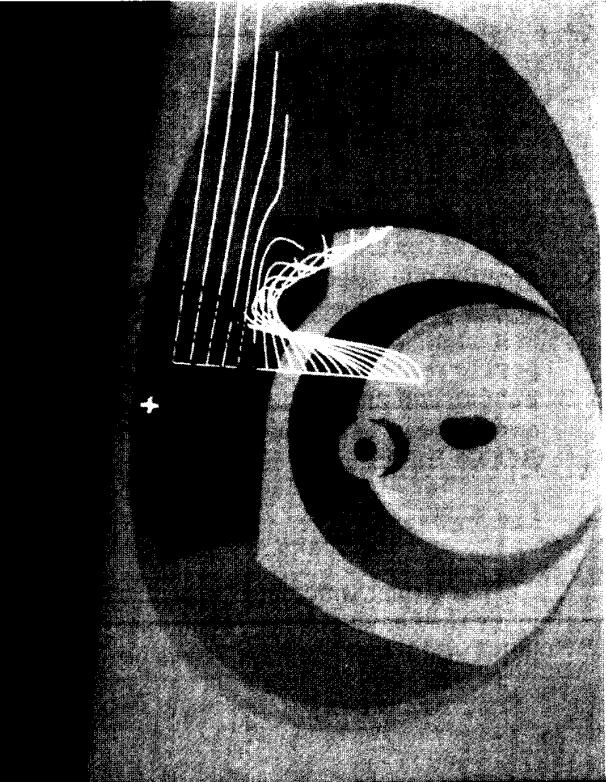


Figure 7: Sphere indicating that the top of the take may be grabbed